This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (currently amended): A method for communicating across a heterogeneous network ~~having components~~ including a sender and a receiver with dissimilar data structure definitions, comprising:

        prefixing an encoded data structure with a length value that reflects ~~the size~~ a size of the encoded data structure, the encoded data structure being transmitted from the sender to the receiver.

2.   (currently amended): The method of Claim 1, further comprising, when the receiver is a down-level receiver, decoding the encoded data structure by the receiver via reading, ~~a down-level receiver reads~~ the length value and ~~decoding the encoded data structure according to~~ based on the receiver's data definition.

3.   (original): The method of Claim 2, further comprising, upon completion of decoding, the receiver determining ~~the amount~~ an amount of the encoded data structure that was decoded and skipping a remainder ~~the remainder~~ of the encoded data structure according to the length value.

4.   (original): The method of Claim 3, wherein the method is implemented through instructions on a computer-readable medium, for communicating data between programs along a data communication path.

5.   (currently amended): The method of Claim 1, further comprising, ~~for~~ if the sender is a down-level sender and the receiver is an up-level receiver, and if the up-level definition of a structure has more data elements than the down-level definition of the structure, for built-in type data fields, automatically assigning a default value to any field for which the received data has provided no value, and, for derived type data fields, calling an initialization routine which assigns a default value to any

3

built-in type data field or calls the initialization routine for a derived type data field.

6.   (original):   The method of Claim 5, wherein the method is implemented through instructions on a computer-readable medium, for communicating data between programs along a data communication path.

7.   (currently amended):   A method for communicating across a heterogeneous network ~~having components~~ including a sender and a receiver with dissimilar data structure definitions, comprising:

    ~~for~~ when the sender is an up-level sender and the receiver is a down-level receiver, if the up-level definition of a structure has more data elements than the down-level definition of the structure, then prefixing an encoded data structure with a length value that reflects ~~the size~~ a size of the encoded data structure and decoding the encoded data structure in which ~~a down-level~~ the down-level receiver reads the length value and ~~decoding~~ decodes the encoded data structure according to the receiver's data definition and upon completion of decoding, the receiver determining ~~the amount~~ an amount of the encoded data structure that was decoded and skipping a remainder ~~the remainder~~ of the encoded data structure according to the length value;

    ~~for~~ when the sender is a down-level sender and the receiver is an up-level receiver, if the up-level definition of a structure has data elements than the down-level definition of the structure, for built-in type data fields, automatically assigning a default value to any field for which the received data has provided no value, and, for derived type data fields, calling an initialization routine which assigns a default value to any built-in type data field or calls the initialization routine for a derived type data field; and

    when the sender is an up-level server and the receiver is a down-level client, if the up-level definition of a structure requires more data elements than the down-level definition of data elements, then following a set of predetermined rules which include extending only data structures which are passed from ~~a server to a client~~ the up-level server to the down-level client; ensuring ~~that a~~ that the down-level client's ignorance of extended data causes no ill effects in that client's operational behavior; allowing down-level clients to interact with up-level servers and disallowing up-level

clients from interacting with down-level servers; and, in cases where extensions are needed for data structures passed from a client to a server, defining a new data structure that includes both old data fields and new data fields.

8.   (original):   The method of Claim 7, wherein the method is implemented through instructions on a computer-readable medium, for communicating data between programs along a data communication path.

9.   (original):   The method of Claim 8, wherein the method is practiced without creating new data structures.

10.   (original):   The method of Claim 8, wherein the method does not use a lock-step migration strategy.

11.   (original):   The method of Claim 8, wherein the built-in type includes at least one of the group consisting of integer, floating point, Boolean, and string.

12.   (original):   The method of Claim 8, wherein the derived type includes at least one of structure and union.

13.   (canceled)

14.   (original):   The method of Claim 8, wherein the method is implemented using a procedure calling model for distributed applications and a standard representation for data in the network to support heterogeneous network.

15.   (original):   The method of Claim 14, wherein the procedure calling model is defined by the Remote Procedure Call (RPC) package and the standard representation of data is accomplished through the External Data Representation (XDR).